



主 主 策  
编 审 划  
吴再陵 军 滨  
沈 王 晓 敏

# 全国青少年信息学 奥林匹克联赛 培训教材 (中学)

全国青少年信息学奥林匹克竞赛培训丛书

南京大学出版社





主编  
主副主策

编编  
审划

吴再陵  
高建君  
沈军  
王晓敏

徐滨

# 全国青少年信息学

# 奥林匹克联赛

# 培训教材 (中学)

全国青少年信息学奥林匹克竞赛培训丛书

南京大学出版社



### 图书在版编目(CIP)数据

全国青少年信息学奥林匹克联赛培训教材. 中学/吴再陵主编. —南京:南京大学出版社, 2002. 8(2009. 1 重印)

ISBN 978 - 7 - 305 - 03823 - 5

I. 全... II. 吴... III. 计算机课—中学—教学参考资料 IV. G634. 673

中国版本图书馆 CIP 数据核字(2002)第047250号

丛 书 名 全国青少年信息学奥林匹克竞赛培训丛书  
书 名 全国青少年信息学奥林匹克联赛培训教材(中学)  
主 编 吴再陵  
出版发行 南京大学出版社  
社 址 南京市汉口路 22 号 邮编 210093  
电 话 025—83596923 025—83592317 传真 025—83328362  
网 址 <http://press.nju.edu.cn>  
电子邮箱 [nupress1@public1.ptt.js.cn](mailto:nupress1@public1.ptt.js.cn)  
经 销 全国各地新华书店  
印 刷 徐州新华印刷厂  
开 本 787×1092 1/16 印张 18.5 字数 445 千  
版 次 2002 年 7 月第 1 版 2009 年 1 月第 15 次印刷  
印 数 79001~82000 册  
ISBN 978-7-305-03823-5  
定 价 28.00 元

---

\* 版权所有,侵权必究

\* 凡购买南大版图书,如有印装质量问题,请与所购图书销售部门联系调换

# 全国青少年信息学奥林匹克竞赛培训丛书

## 编 委 会

**顾 问:** 吴文虎 宋秀芳 杜子德

**主任委员:** 周稽裘 吴国彬

**副主任委员:** 陆志平 李树奎 李立新 蔡绍稷  
冯少东 王晓敏 顾其兵

**委 员:** 徐 滨 郁建文 沈 军 章维铨  
曹 文 吴再陵 林厚从 林 晖  
高建君 朱 驯 金 艳 赵 聆  
王 静 王 超 许 冰 谷爱清  
李 玮

# 序 言

江泽民主席在给第 12 届国际信息学奥林匹克竞赛的贺信中指出：“在人类即将进入新世纪之际，以信息科技和生命科技为核心的科技进步与创新，正在深刻地改变着人类的生产和生活方式，推动着世界文明的发展。青年是人类的未来，也是世界科技发展的未来。国际信息学奥林匹克竞赛活动，对年轻一代了解和掌握现代科学技术，养成创新精神，具有重要作用。”

国际信息学奥林匹克(International Olympiad in Informatics)，简称 IOI。从 1989 年到 2001 年，13 年赛事的健康发展得益于联合国教科文组织(UNESCO)为这项赛事所做的准确定位：通过竞赛形式对有才华的青少年起到激励作用，促其能力得以发展；让青少年彼此建立联系，推动经验交流，给学校这一类课程增加活力；建立起教育工作者与专家档次上的国际联系，推进学术思想的交流。概括起来说，就是：启迪思路，激励英才，发展学科，促进交流。

作为与国际学科奥林匹克竞赛接轨的我国青少年信息学奥林匹克竞赛，其宗旨就是在青少年中普及信息科学，用最先进的智力工具来开发未来建设者的智力，让他们满怀信心地跃上信息平台。在我国，每年国家集训队都要将“怎样做人，怎样做事，怎样求知和怎样健体”的指导思想纳入培训计划。13 年来中国队共派出参赛选手 51 人次，累计获金牌 26 块、银牌 14 块、铜牌 11 块、届届名列前茅，正是因为坚持了全面素质教育的指导思想，把造就高素质有创造精神的人才作为活动的定位目标。

学科奥林匹克是智力与能力的竞赛，注重考查全面素质与创造能力。从这个意义上讲，信息学奥林匹克活动是素质教育的一个大课堂。回顾 13 年赛事可以看出，参加高手云集的这种世界大赛是有相当难度的：

- (1) 没有大纲，赛题范围没有界定，谁也无法去猜测每年的主办国会出什么类型的难题；
- (2) 计算机科学与技术发展很快，层出不穷的新思路和新成果会反映到试题中来；
- (3) 所要解决的试题往往涉及图论、组合数学、人工智能等大学开设的课程知识；
- (4) 比较短的给定解题时间与刁难的测试数据让选手必须拿出高超和精巧的解法，无论在时间上还是空间上都是优化的解法才能取得高分。有许多赛题没有固定的现成的解法，选手要在比赛现场凭借实力，理出思路，构建数学模型，写出算法，编出程序，运行并验证整个构思是否正确，出解的时间是否能达到题目的要求，等等。

可以看出，在这一过程中最重要的是创造能力，我们为激发创新精神，培养创造能力，需要树立新的教育观念和教学方法，还要利用现代化的教学手段。引导学生学用电脑，在使用中帮助开发人脑，这可能是信息学奥林匹克活动的最重要的一个特点。我认为在这项活动中应该培养学生的四种能力：自学能力；实践动手能力；创新能力；上网获取知识，并能区分有用知识和无用知识的能力。这样做的结果使许多选手不但有能力在世界赛场上拿金牌，也有能力在学校的学习中名列前茅。

信息学奥林匹克涌现出一大批出类拔萃的计算机后备人才，在他们的带动下，我国的青少年在普及计算机的大潮中阔步前进，取得了可喜的成绩。历史已雄辩地证明：计算机的普及就是要从娃娃做起，这是“科教兴国”，中华崛起的需要。为了进一步推动普及，在教育部、科协有



关部门的领导下,从 1995 年起江苏省青少年科技中心受中国计算机学会的委托,已连续 7 年成功组织承办了全国的奥林匹克联赛(简称 NOIP)活动,数以万计的青少年从中受益。在这 7 年中,参与这项活动的老师与专家积累了许多宝贵经验,从 1999 年起陆续撰写出版了一套青少年信息学(计算机)奥林匹克竞赛丛书,包含《信息学(计算机)奥林匹克初级本》《信息学(计算机)奥林匹克中级本》《信息学(计算机)奥林匹克高级本》及《全国青少年信息学(计算机)奥林匹克分区联赛试题解析》。该套丛书注重了系统性、入门性与实用性,始终围绕编程实践,以算法分析为主线,讲思想、讲方法,侧重基础训练,引导学生在参与的实践中掌握科学思维方法,提高使用计算机的能力。现根据广大读者的建议与活动普及与发展的需要,将这套丛书中的《信息学(计算机)奥林匹克初级本》与《信息学(计算机)奥林匹克中级本》重新进行了编写,修改后的这两本书可作为广大青少年参与信息学奥林匹克活动的培训教材,我相信这一定会对信息技术的普及起到推波助澜的作用。

青少年是国家的希望,大力开展青少年计算机普及教育活动,必将为我国 21 世纪在科技、经济、文化各个领域的腾飞,准备好一支用现代高科技武装起来的建设者队伍。在诸多因素中,人是决定因素,不断提高青少年的科学素养是中华民族永远昂首屹立在世界东方的根基所在。“精心育桃李,切望青胜蓝”是我和编写这套丛书的老师们的共同心愿。

国际信息学奥林匹克中国队总教练  
清华大学计算机系教授,博士生导师

2002. 4. 15

# 前 言

全国青少年信息学奥林匹克竞赛及其联赛活动是经中国科协、教育部批准的一项全国性青少年学科竞赛活动。它吸引了许多青少年学习计算机基本知识,学习程序设计的基本方法,积极参与各级各类的竞赛活动。为了使广大读者能尽快地学习和掌握有关程序设计的基本知识和方法,学习简单常用的数据组织与数据处理的方法和思想,我们组织编写了这一套教材。

程序设计是一种创造性劳动,其本身涉及三个方面的内容:计算机语言、开发环境与应用(设计开发)。三者既相对独立又相互联系,特别是前两者与后者的关系更是难以把握,因为语言的学习是为了应用,但学好语言并不意味着学好应用。然而,语言的好坏需要通过其表达能力(即应用)来体现。因此,如何处理好语言与应用、语言与开发环境之间的关系是十分重要的。另一方面,为了学好应用(程序设计),除了语言、环境的知识外,对应用本身的一些规律也必须学习,学习经典的、习惯性的、经过实践证明有效的一些基本模式,学习各种基本模式的逻辑组合规则,以此展开自己的思维,针对具体问题进行灵活应用。Pascal 语言作为过程模型程序设计的代表,有其独特的魅力,作为程序设计的起点,其强调的结构化设计方法和思想蕴涵着丰富的哲理,是培养程序设计思维的良好入门。

教材是一种教学思想和理念的反映,教材的体系反映了一定的教学思路。本教材特别注重应用模式的发掘和规范,力求将模式及其建构的思想与用具体语言来描写这两方面的关系阐述清楚。第一章介绍了算法的概念及其描述方法。第二章介绍了 Turbo Pascal 7.0 开发环境的基本知识、功能与基本使用方法。第三章至第十章介绍了 Pascal 语言的各种基本知识,并融合了各种基本应用模式,以体现语言本身的描写方法和描写能力。特别是第四章介绍了控制逻辑的三种组合模式及其语言描述、第七章介绍了子程序及其耦合方法与如何用语言描述,它们是结构化程序设计的精髓。第十一章简要介绍了面向对象的程序设计方法和思想,强调了接口和实现的分离,有利于大规模应用的协同开发和维护。第十二章从整体角度,进一步归纳了若干应用模式,分析其模式规律。附录给出 Turbo Pascal 7.0 的相关资料。

本书第一、三章由倪诗律编写,第二、五、七章由高建君编写,第四章由林晖编写,第六章由王静编写,第八、九章由林厚从编写,第十章由许冰编写,第十一、十二章由吴再陵编写。全书由沈军主审。本书在编写过程中,得到江苏省青少年计算机教育活动中心的王晓敏、赵聆老师的大力帮助和支持,在此表示由衷的感谢。

由于编者的水平有限,加之时间仓促,书中不当和疏漏之处,在所难免。特别是关于基本模式的分析和建构策略的讨论,未能展开和深入,只是引入了基于模式及其建构的教学思路,思路的具体实现和执行还有待于读者自己的体会。在此,恳请广大读者批评指正。

沈 军

东南大学计算机系

江苏省青少年信息学奥林匹克竞赛委员会普及委员会

2002年4月25日

# 目 录

全国青少年信息学奥林匹克联赛培训教材(中学)

序 言 .....	吴文虎
前 言 .....	沈 军
<b>第一章 算法及算法的描述 .....</b>	<b>( 1 )</b>
1.1 算法知识基础 .....	( 1 )
1.2 算法的描述方法 .....	( 1 )
1.2.1 自然语言 .....	( 1 )
1.2.2 N-S图 .....	( 2 )
1.2.3 程序设计语言 .....	( 4 )
1.3 算法的实现 .....	( 5 )
1.3.1 算法的实现 .....	( 5 )
<b>第二章 Turbo Pascal 7.0 基础 .....</b>	<b>( 7 )</b>
2.1 Turbo Pascal 简介 .....	( 7 )
2.2 安装及启动 .....	( 7 )
2.3 集成环境及菜单使用 .....	( 9 )
2.4 程序的输入和调试 .....	( 15 )
<b>第三章 Pascal 程序设计语言基础 .....</b>	<b>( 18 )</b>
3.1 概述 .....	( 18 )
3.1.1 Pascal 语言特点 .....	( 18 )
3.1.2 Pascal 程序结构 .....	( 18 )
3.1.3 基本符号、保留字和标识符 .....	( 20 )
3.2 数据类型、常量、变量及说明方法 .....	( 21 )
3.2.1 标准数据类型 .....	( 21 )
3.2.2 常量 .....	( 23 )
3.2.3 变量 .....	( 23 )
3.3 函数与表达式 .....	( 24 )
3.3.1 Pascal 标准函数 .....	( 24 )
3.3.2 Pascal 的运算符及表达式 .....	( 25 )
<b>第四章 程序设计初步 .....</b>	<b>( 27 )</b>
4.1 顺序结构的程序设计 .....	( 27 )
4.1.1 赋值语句 .....	( 28 )
4.1.2 输入语句(read, readln) .....	( 29 )
4.1.3 输出语句(write, writeln) .....	( 32 )





4.1.4	复合语句	(35)
4.1.5	综合应用	(35)
4.2	选择结构的程序设计	(38)
4.2.1	if 语句	(38)
4.2.2	if 语句的嵌套	(43)
4.2.3	case 语句	(45)
4.2.4	综合应用	(46)
4.3	循环结构的程序设计	(49)
4.3.1	计数循环(for/to/do 语句)	(49)
4.3.2	当型循环(while/do 语句)	(52)
4.3.3	直到型循环(repeat/until 语句)	(55)
4.3.4	多重循环结构	(57)
4.3.5	关于 goto 语句	(59)
4.3.6	综合应用	(61)
<b>第五章</b>	<b>枚举类型和子界类型</b>	<b>(67)</b>
5.1	枚举类型	(67)
5.1.1	枚举类型的定义	(67)
5.1.2	枚举类型数据的特点及应用	(68)
5.2	子界类型	(71)
5.2.1	子界类型的定义	(71)
5.2.2	子界类型数据的特点及应用	(73)
5.3	类型相容及应用	(75)
<b>第六章</b>	<b>数组</b>	<b>(80)</b>
6.1	数组的概念和定义	(80)
6.2	一维数组	(82)
6.2.1	一维数组的定义及数据元素引用	(82)
6.2.2	一维数组的基本操作	(83)
6.2.3	一维数组的应用举例	(89)
6.3	多维数组	(93)
6.3.1	多维数组定义及数据元素的引用	(93)
6.3.2	多维数组的应用	(94)
6.4	字符数组及字符串	(99)
6.4.1	紧缩数组	(99)
6.4.2	字符数组与字符串	(99)
<b>第七章</b>	<b>过程和函数</b>	<b>(105)</b>
7.1	子程序的概念	(105)
7.2	函数定义及调用	(105)
7.2.1	Pascal 标准函数	(105)
7.2.2	用户自定义函数	(106)



7.2.3	函数调用 .....	(107)
7.3	过程定义及调用 .....	(109)
7.3.1	标准过程 .....	(109)
7.3.2	用户自定义过程 .....	(109)
7.3.3	过程调用 .....	(111)
7.4	变量及其作用域 .....	(113)
7.5	参数的传递 .....	(114)
7.5.1	形参和实参 .....	(114)
7.5.2	值参和变参 .....	(115)
7.6	程序的嵌套和递归调用 .....	(121)
7.6.1	子程序的嵌套 .....	(121)
7.6.2	提前引用子程序 .....	(123)
7.6.3	递归调用 .....	(125)
7.7	综合应用 .....	(130)
<b>第八章</b>	<b>集合和记录</b> .....	(135)
8.1	集 合 .....	(135)
8.1.1	集合的概念 .....	(135)
8.1.2	集合类型的定义及集体变量的说明 .....	(135)
8.1.3	集合的赋值 .....	(136)
8.1.4	集合运算的实现 .....	(136)
8.1.5	综合应用 .....	(138)
8.2	记 录 .....	(143)
8.2.1	记录的概念 .....	(143)
8.2.2	记录类型的定义及记录变量的说明 .....	(143)
8.2.3	记录成员的引用 .....	(145)
8.2.4	记录数组 .....	(147)
8.2.5	变体记录 .....	(147)
8.3	综合应用 .....	(148)
<b>第九章</b>	<b>文 件</b> .....	(154)
9.1	文件概述 .....	(154)
9.1.1	文件的概念及特点 .....	(154)
9.1.2	文件的分类 .....	(155)
9.1.3	文件处理的标准过程和标准函数 .....	(155)
9.1.4	文件操作的一般步骤 .....	(156)
9.2	文件的操作及应用 .....	(156)
9.2.1	文本文件的操作步骤 .....	(156)
9.2.2	文本文件的应用举例 .....	(158)
9.2.3	类型文件的操作步骤 .....	(164)
9.2.4	类型文件的应用举例 .....	(165)



<b>第十章 指 针</b> .....	(169)
10.1 静态存储与动态存储.....	(169)
10.2 指针变量及基本使用.....	(170)
10.2.1 指针变量定义.....	(170)
10.2.2 指针变量的基本使用方法.....	(170)
10.3 线性链表.....	(173)
10.3.1 线性链表的概念.....	(173)
10.3.2 线性链表的建立.....	(174)
10.3.3 线性链表的遍历与输出.....	(175)
10.3.4 线性链表的查找.....	(176)
10.3.5 线性链表结点的插入.....	(176)
10.3.6 线性链表结点的删除.....	(179)
10.4 综合应用.....	(180)
10.4.1 求线性表长度的运算.....	(180)
10.4.2 线性链表的排序.....	(181)
10.4.3 线性表的归并算法.....	(183)
10.4.4 循环链表.....	(186)
10.4.5 二叉树.....	(187)
<b>第十一章 单元及面向对象的程序设计初步</b> .....	(189)
11.1 单元及程序设计.....	(189)
11.1.1 单元的概述.....	(189)
11.1.2 单元的使用说明.....	(189)
11.1.3 标准库单元.....	(191)
11.1.4 单元的结构.....	(192)
11.1.5 单元的建立和应用.....	(193)
11.2 面向对象的程序设计.....	(197)
11.2.1 什么是面向对象的程序设计.....	(197)
11.2.2 如何设计.....	(198)
11.2.3 综合应用.....	(201)
<b>第十二章 常用算法介绍</b> .....	(205)
12.1 穷举法.....	(205)
12.2 不同进制数的转换及应用.....	(212)
12.3 高精度计算.....	(219)
12.4 数据排序.....	(225)
12.5 排列和组合.....	(235)
12.6 递推与递归算法深入.....	(245)
12.6.1 递推程序设计.....	(245)
12.6.2 递归算法深入与递归转化为非递归.....	(247)
12.7 回溯算法.....	(255)



12.8 算法评价.....	(268)
附 录.....	(272)
附录一 常用字符的 ASCII 码对照表 .....	(272)
附录二 Turbo Pascal 的保留字 .....	(273)
附录三 Turbo Pascal 的预定义标识符 .....	(273)
附录四 Turbo Pascal 的标准函数 .....	(274)
附录五 Turbo Pascal 的运算符 .....	(274)
附录六 Turbo Pascal 编译、运行过程中的出错信息 .....	(275)
附录七 Turbo Pascal 标准库单元中的过程和函数 .....	(278)

# 第一章 算法及算法的描述

## 1.1 算法知识基础

要用计算机解决一个较为复杂的实际问题,一般要经历如下四个步骤:

- (1) 分析问题:如实际问题能数学化的,则把实际问题抽象为一个带有一般性的数学问题,即为该问题建立一个良好的数学模型;
- (2) 根据分析,并利用分析后确定的数学模型,设计其求解的方法和步骤,即算法设计;
- (3) 用某种形式,如图形、自然语言或类计算机语言来表达已经设计好的算法;
- (4) 用一种计算机语言实现算法,并在计算机上编辑、调试和测试编制好的程序,直到正确满足问题的要求。

从上面的求解问题过程可以看出,关键在于前三步是怎么解决的,第一步其实就是解决模型的数据结构,第二步即为解决问题的算法,第三步即为形式化地描写算法。

所谓“算法”,就是一组有穷的规则,它们规定了解决某一特定类型问题的一系列运算。通俗地讲,是指完成某一项任务的方法和步骤,对程序设计而言是对解题过程的准确而完整的描述,它可以是数学公式也可以是规则,但最终它必须表示成上机操作运行的命令。从计算机系统的角度,算法也可理解为:程序(数据处理)+ 数据结构(数据组织)。

算法具有如下的性质:

- (1) 有限性,算法中的运算项是有限的,且每个运算项都可在有限的时间内完成;
- (2) 确定性,即算法中的每一项运算都有明确的定义,无二义性;
- (3) 输入输出,可以没有输入项,但一定要有输出运算项;
- (4) 可行性,即对于任意给定的合法输入均可得到相应的正确的输出结果。

常见的算法有:穷举法,迭代法,递推法、递归法、回溯法、深度及广度搜索法、动态归划、构造法等等。

## 1.2 算法的描述方法

为了描述一个算法,可以采用多种不同的方法,常用的有自然语言、程序流程图、N-S图、类程序设计语言以及程序设计语言等方法。

### 1.2.1 自然语言

自然语言描述算法也就是用人们日常使用的语言来描述,该描述方法通俗易懂,但也有一些缺点:首先是文字描述比较繁琐,如“把5给变量A”不如写成“ $5 \rightarrow A$ ”;其次,自然语言容易出现二义性问题,如“张三要李四把他的笔借给他”,这两个“他”的含义不明确。另外,用自然语言描述判断和转移时不那么直观清晰。因此,我们只在学习程序设计的初期或口头交流算



法思想的时候使用自然语言的描述。

下面我们通过实例来讲述用自然语言描述算法。

**【例 1-1】** 交换 A 和 B 的值。

- (1) 赋 A 和 B 的初值;
- (2) 把 A 的值给 X;
- (3) 把 B 的值给 A;
- (4) 把 X 的值给 B;
- (5) 输出 A 和 B 的值。

**【例 1-2】** 将 A 和 B 中较大的值给 MAX。

- (1) 赋 A 和 B 的初值;
- (2) 将 A 的值给 MAX;
- (3) 比较  $B > MAX$ , 如果条件成立, 则将 B 的值给 MAX;
- (4) 输出 MAX 的值。

**【例 1-3】** 给定一个正整数 N, 判断它是否为素数。

素数的特征为除 1 和该数本身之外, 不能被其他任何整数整除。因此我们算法的基本思想为: 将 N 被 2、3、……、N-1 除, 如果都除不尽, 则 N 为素数。根据该思路得到的算法如下:

- (1) 设除数为 I, I 的值从 2 变化到 N-1, I 的初值为 2;
- (2) 用 I 除 N, 得到余数为 R;
- (3) 如果  $R=0$ , 则表示 N 能被 I 整除, N 不是素数, 算法结束; 否则, 表示 N 不能被 I 整除, 可能为素数, 继续;
- (4) 使 I 的值加上 1;
- (5) 如果  $I \leq N-1$ , 那么返回(2); 否则, 表示 N 已被 2 到 N-1 除, 且都不能被整除, 可以判断 N 为素数, 算法结束。

### 1.2.2 N-S 图

1973 年, 美国学者 I. Nassi 和 B. Shneiderman 提出了一种在流程图中完全去掉流程线, 全部算法写在一个矩形框内, 在框内还可以包含其他框的流程图形式。即由一些基本的框组成一个大的框, 这种流程图又称 N-S 结构流程图(以两个人的名字的头一个字母组成)。N-S 图包括顺序、选择和循环等三种基本结构。

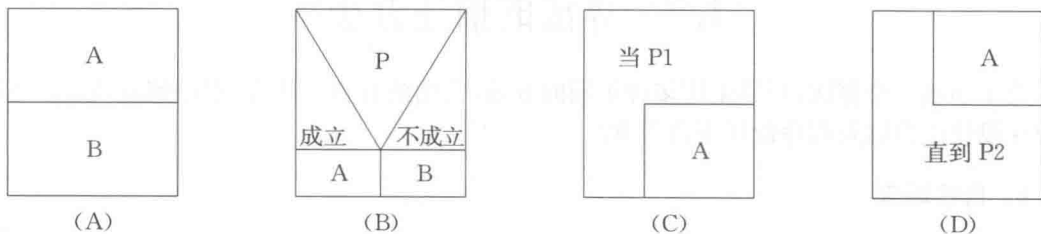


图 1-1 N-S 图表示的 3 种基本结构的基本元素框

图 1-1(A) 表示顺序结构, 也就是执行完操作 A 后就立即执行操作 B。图 1-1(B) 表示选择结构, 也就是当条件 P 成立时执行操作 A, 否则执行操作 B, 选择结构也称为分支结构。



图 1-1(C)表示 WHILE(当)型循环结构,表示“当条件 P1 成立时反复执行操作 A”。图 1-1(D)表示 UNTIL(直到)型循环结构,表示“反复执行操作 A 直到条件 P2 成立为止”。

上面三个程序,用 N-S 图表示,可表示如下:

**【例 1-4】** 交换 A 和 B 的值。

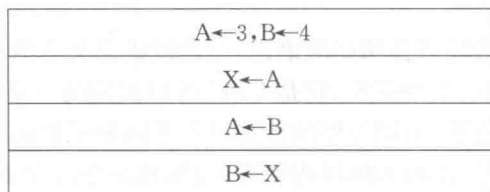


图 1-2

**【例 1-5】** 将 A 和 B 中较大的值给 MAX。

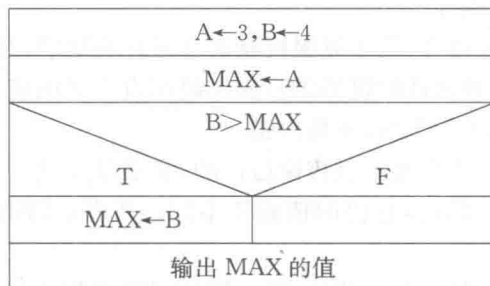


图 1-3

**【例 1-6】** 给定一个正整数 N,判断它是否为素数。

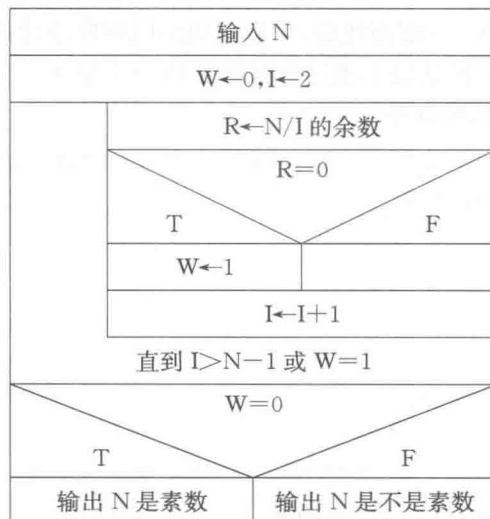


图 1-4



N-S图的精髓在于,几种基本图形可以任意嵌套,即A图形中可以是B图形,而B图形中又可以是A图形……,尽管基本图形只有四种,但其变化是无穷的。怎么变则取决于人的思维。可见,N-S图实际上反映的是一种逻辑思维,蕴涵着深层哲理。

### 1.2.3 程序设计语言

计算机中的语言分为低级语言和高级语言,而低级语言又分为机器语言和汇编语言两种。

机器语言是一种CPU的指令系统,它是CPU可以识别的一系列由0和1这种二进制代码组成的指令。它依赖于机器,不同类型的计算机有不同的机器语言,机器语言的程序由许多机器指令组成,每条指令由操作码和地址码组成,数据和指令放在不同的地址单元中。因此由机器语言编制的程序就是目标程序,无需经过翻译就可以由计算机来运行,它的执行效率较高。但由于机器语言完全由0、1组成,即使是专业人员仍感难写、难读、难修改,程序的生产效率很低,质量难以保证,只有少数的计算机专业人员能掌握,再由于各种机器的指令系统不一样,编制出来的程序没有通用性。

汇编语言它是一种符号语言,为了克服机器语言固有的缺陷,20世纪50年代中期出现了“汇编语言”,它将难以记忆和辨认的机器语言操作码用有意义的英文单词作为“助记符”来代替0、1进行编程,这样看起来较直观,不易出错。

但就机器而言汇编语言指令是不能直接运行的,它必须经过一个汇编程序编译成由机器语言组成的目标程序后才能执行,它仍旧依赖于不同的机器,不同的计算机有不同的汇编语言,不能通用。

机器语言和汇编语言均是面向机器的,用它们编制程序都受到限制,既费工又费时,效率很低。为了从根本上解决问题,1954年出现了Fortran语言及其他一些程序设计语言,这些语言均与机器无关,而且与人的自然语言比较接近,我们统称为高级语言。

常见的高级语言还有Cobol,Pascal,Basic,C,C++、Java等等。

当一个问题的算法用N-S图描述后,可以很快地以程序设计语言来描述,用程序设计语言描述的算法即是程序。下面就是上述三个例题的Pascal程序。

**【例1-7】** 交换a和b的值的程序。

```
program pl_1(input,output);
var a,b,x:integer;
begin
  a:=3;b:=4;
  x:=a;a:=b;b:=x;
  writeln('a=',a,'b=',b);
end.
```

**【例1-8】** 将a和b中大的值给max的程序。





```

program p1_2(input,output);
var a,b,max:integer;
begin
  a:=3;b:=4;
  max:=a;
  if b>max then max:=b;
  writeln('max=',max);
end.

```

**【例 1-9】** 给定一个正整数  $n$ , 判断它是否为素数的程序。

```

program p1_3(input,output);
var i,n,r,w:integer;
begin
  write('n=');readln(n);
  w:=0;i:=2;
  repeat
    r:=n mod i;
    if r=0 then w:=1;
    i:=i+1;
  until (i>n-1) or (w=1);
  if w=0 then writeln('yes') else writeln('no')
end.

```

## 1.3 算法的实现

### 1.3.1 算法的实现

当我们拿到一个具体问题时,我们首先应该确定该问题的算法,根据算法写出程序执行的框图,在我们学习 Pascal 语言以后,我们就能根据框图写出程序,下面我们就具体问题谈谈算法的实现。

**【例 1-10】** 求出  $1+2+3+\dots+N$  的值。

**【问题分析】** 根据题目,我们用  $S$  表示所求数的和,用  $T$  表示加数,这样我们就可以让  $T$  从 1 变化到  $N$ ,每次让  $T$  增加 1,然后将  $T$  不断地加入到  $S$  中,就可以求出  $S$  的值。用自然语言描述的算法具体表示如下:

第 1 步:置初值,即  $S=0, T=1$ ;

第 2 步:使  $S=S+T$ ,所求出的和仍放入  $S$  中;